

UNIVERSIDAD NACIONAL DEL SUR
BAHIA BLANCA

1
5

DEPARTAMENTO DE CIENCIAS E INGENIERÍA DE LA COMPUTACIÓN

PRINCIPIOS Y HERRAMIENTAS DE
PROGRAMACION

CÓDIGO: 7887

ÁREA N°:

CARRERAS Y PLANES

PROFESOR RESPONSABLE:

Dr. Ignacio Ponzoni – Dra. Jessica Carballido

CARGA HORARIA	Teoría	Práctica	Laboratorio	CANTIDAD DE SEMANAS	
	64	64	(todas)		16

CORRELATIVAS

PARA CURSAR LA MATERIA		PARA APROBAR LA MATERIA	
APROBADAS	CURSADAS	APROBADAS	CURSADAS

DESCRIPCIÓN

El objetivo principal de la materia es que los alumnos adquieran la capacidad de desarrollar programas para resolver problemas de pequeña escala. El desarrollo de un programa se concibe como un proceso que abarca varias etapas:

- *La interpretación adecuada del enunciado a través del cual se plantea el problema.*
- *El diseño de un algoritmo que modela la resolución del problema.*
- *La implementación del algoritmo en un lenguaje de programación imperativo.*
- *La verificación de la solución.*

En el desarrollo del curso se pone especial énfasis en que los alumnos puedan reconocer y desarrollar cada una de las etapas mencionadas, y al resolver problemas puedan destinar a cada etapa el esfuerzo que merece en cada caso. Se introducen diferentes estrategias para resolver problemas, como dividir un problema en subproblemas; obtener la solución a través de refinamientos sucesivos; particularizar y generalizar; y la elección adecuada de ejemplos y casos de prueba. En la etapa de implementación se introducen criterios que faciliten la legibilidad del programa y su mantenimiento. En todas las etapas se aplica y refuerza la capacidad de abstracción de los alumnos. El diseño del algoritmo y su implementación se concentra en seleccionar las estructuras de control adecuadas ya que los datos pueden ser representados usando tipos elementales. También es un objetivo de la materia que los alumnos aprendan los conceptos y terminología básicos de la disciplina.

Competencias previas y a desarrollar

La concreción del objetivo de la materia implica el aprendizaje de contenidos conceptuales, actitudinales y procedimentales. Se requiere que los alumnos:

- Interpreten enunciados.
- Diseñen algoritmos aplicando los criterios de programación estructurada, modularización

DEPARTAMENTO DE CIENCIAS E INGENIERÍA DE LA COMPUTACIÓN

PRINCIPIOS Y HERRAMIENTAS DE
PROGRAMACION

CÓDIGO: 7887

ÁREA N°:

y refinamiento paso a paso.

- Implementen soluciones en C y R utilizando adecuadamente las estructuras de control.
- Verifiquen la solución implementada en lenguaje de programación respecto al enunciado para un conjunto significativo de casos de prueba.
- Documenten el código para favorecer la legibilidad.
- Conozcan y apliquen adecuadamente los conceptos de variable, expresión, instrucción, tipo de dato, algoritmo, programa, subprograma, parámetro.

METODOLOGÍA DE ENSEÑANZA

Las actividades se realizan de una manera expositiva en las clases teóricas, con la participación de los alumnos en la resolución de los problemas. Luego, en la medida en que se van aprendiendo los lenguajes de programación, se cambia a un modo en el que los alumnos pueden ir probando lo que se da en teoría.

En la práctica trabajan con los conceptos aprendidos, diseñando e implementando ya sea en papel (los algoritmos) o en máquina (durante la fase que aprenden C y R).

En todos los casos las clases se dictan en laboratorio.

El programa de la materia, los trabajos prácticos, el mecanismo y cronograma de evaluación están disponibles en la página web de la materia.

MECANISMO DE EVALUACIÓN

El mecanismo de evaluación para el cursado consiste en un examen en máquina en el cual se evalúa la resolución de varios problemas de pequeña escala en los dos lenguajes de programación abordados durante la cursada.

PROGRAMA SINTÉTICO

Algoritmia

- Diseño de algoritmos
- Conceptos básicos de complejidad

Programación en Lenguaje Imperativo

- Compilación
- Sistemas de tipos
- Tipado estático
- Conversiones explícitas e implícitas
- Caso de estudio: C
- Sintaxis
- Estructura de un programa en C
- Tipos básicos - Modificadores de tipo
 - Operadores aritméticos, de asignación, relacionales, de indirección
 - Conversión de tipos
 - Sentencias condicionales
- Sentencias para estructuras iterativas
- Funciones

Programación en Lenguaje Script

Lenguajes compilados vs. Lenguajes interpretados

Sistema de tipos

Tipado dinámico

Conversiones explícitas e implícitas

Caso de estudio: R

Sintaxis

Características básicas del lenguaje

Operadores aritméticos y lógicos

Sentencias para estructuras iterativas y condicionales

Tipos de datos predefinidos y sus operaciones básicas

Declaración de funciones

Ligadura dinámica y conversión de tipos

Funciones y características avanzadas

Generación de gráficos y reportes estandarizados

PROGRAMA ANALÍTICO

Algoritmia

Diseño de algoritmos

Algoritmos cotidianos

Algoritmos, datos e instrucciones

Expresiones aritméticas y lógicas

Asignación

Estructuras de control condicionales

Si ... entonces ...

En caso de ... hacer ...

Estructuras de control iterativas

Repetir mientras ... hacer ...

Repetir ... hasta ...

Repetir ... veces ...

Primitivas.

Conceptos básicos de complejidad

Tiempos de ejecución de algoritmos simples

Programación en Lenguaje Imperativo

Compilación

Sistemas de tipos

Tipado estático

Conversiones explícitas e implícitas

Caso de estudio: C

Sintaxis. Diagramas sintácticos

Estructura de un programa en C

Función main()

Delimitadores de bloque

Variables locales al bloque
Variables globales
Constantes
Edición, compilación, depuración
Tipos básicos - Modificadores de tipo
Operadores aritméticos, de asignación, relacionales, de indirección (concepto de puntero)
Conversión de tipos
Sentencias condicionales: If ... Else, Swich
Sentencias para estructuras iterativas: While, Do... while, For, Break, Continue
Declaración de Funciones
Definición (implementación) – parámetros formales
Llamada a funciones – parámetros actuales
Parámetros por valor y parámetros por referencia
Entrada/Salida estándar

Programación en Lenguaje Script

Lenguajes compilados vs. Lenguajes interpretados

Sistema de tipos

Tipado dinámico

Conversiones explícitas e implícitas

Caso de estudio: R

Características básicas del lenguaje

Creación, asignación y visualización de variables

Operadores aritméticos y lógicos

Sentencias para estructuras iterativas y condicionales

Tipos de datos predefinidos y sus operaciones básicas

String

Vector

Fact

Matrix

List

Dataframe

Declaración de funciones

Ligadura dinámica y conversión de tipos

Operaciones y formatos de Entrada/Salida de datos

Características avanzadas. Funciones *apply*, *any*, *all*, y *rep*

Generación de gráficos y reportes estandarizados. Funciones *plot*, *par*, *abline* y *points*

UNIVERSIDAD NACIONAL DEL SUR
BAHIA BLANCA

5

5

DEPARTAMENTO DE CIENCIAS E INGENIERÍA DE LA COMPUTACIÓN

PRINCIPIOS Y HERRAMIENTAS DE
PROGRAMACION

CÓDIGO: 7887

ÁREA N°:

BIBLIOGRAFIA

- Fundamentos de Algoritmia, G. Brassard y P. Bratley. (ISBN 848966000X)
- Introduction to Algorithms (2nd ed), Cormen, T. H., Leiserson, C. E., Rivest, R. L. y Stein, C.
- Introduction to Algorithms. A Creative Approach, Mamber, U.
- The Design and Analysis of Computer Algorithms, Aho, A.
- The *C Programming Language (2nd Edition)*, Kernighan, Brian W. y Ritchie, Dennis M.: Prentice Hall, 1988.
- The Art of Computer Programming, Knuth, D. E.
- Algorithms in C (3r ed), Sedgewick, R.
- <http://www.r-project.org/> (The R manuals)

AÑO

2013

FIRMA PROFESOR RESPONSABLE

VISADO

COORDINADOR AREA

SECRETARIO ACADÉMICO

DIRECTOR
DEPARTAMENTO